Sensor Networks With Secure Public-Key over GF (2^m)

Xu Huang
Faculty of Information Sciences and Engineering, University of Canberra, ACT 2601, Australia
Xu.Huang@canberra.edu.au

Abstract—**In Observing and interacting with the real physical world sensor networks now take new opportunities for play the roles in almost everywhere. They are composed of a large number of sensor nodes and each sensor node has capabilities of sensing, processing, and communication, which underpinned by limited energy. Reducing energy consumption was, is and will be a major concern in sensor networks. For the given extremely limited hardware resources on sensor nodes and the inclement deploying environment, the adversary attack becomes a serious security threat toward wireless sensor networks. In this paper a secure public-key based effective and efficient sensor network is to be introduced. A saving time, reducing computing energy, algorithm is carefully investigated. The powerful elliptic curve cryptography (ECC) over GF (2^m) is also investigated with hidden generator point.**

Keywords-elliptic curve crytographic;public key; fast calculations in ECC, complementary recoding, 1's complement of binary numbers, hidden generator point, denial-of-service attack, man-in-middle attack.

I. INTRODUCTION

The advancements in sensor technology, several sensor networks have been being deployed in various applications for observing and interacting with the physical world. Those sensor networks are composed of a large number of sensor nodes and each sensor node has capabilities of sensing, processing, and communication. These sensor nodes are normally deployed in environments where they may be hard to access and provide various useful data. It is well known that applications for wireless sensor networks fall in three major categories, namely (1) periodic sensing; (2) event-driven; and (3) query-based. However, all those categories are challenges because of the limited battery resources on each sensor node. The nodes are usually deployed in an unattended manner. It is not easy work to replace the batteries. Therefore, reducing energy consumption was, is and will be a major concern in sensor networks.

For the given the extremely limited hardware resources on sensor nodes and the inclement deploying environment, the adversary attack becomes a serious security threat toward wireless sensor networks. Without adequate defenses mechanism, the adversary can simply inundate the network by flooding the bogus data packets, and paralyze the partial or whole sensor network by depleting mode battery power.

There are many ways to discuss reducing energy consumption such as reducing the communication using the temporal and/or spatial correlation of sensor readings [1-4], decreasing the computing time [5-12] within the security systems, and etc.

It is noted that recently, there is a trend for the sensor networks that the sensor group leaders rather than sensors communicate to the end database as there is an effect way to save the storages and power energy for the "pre-computing data" in the sensor networks [13-15]. However, the algorithms related to this trend will inherently create chance for an attacker to make a "man-in-middle" attack, which is shown in Figure 1.

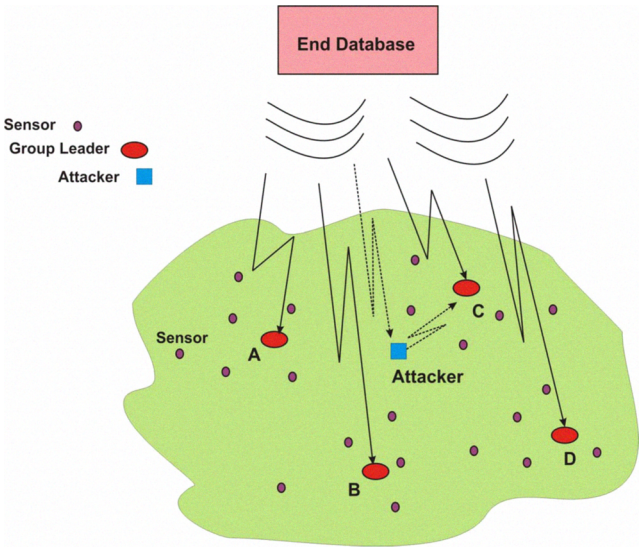

Figure 1. Schematic diagram shows possible attack made by "man-in-middle" in sensor network (in particular for the case with "group leader" communications.)

Therefore there is necessary to design security system to protect the networks from the man-in-middle attacks. In this paper we are going to present two ways to protect the network from man-in-middle attacks based on hidden generator point over elliptic curve cryptography for public keys.

The next section will discuss the general tradition elliptic curve cryptography with our focus, which will be modified for our proposed protocols to protect the networks from the man-in-middle attacks. In section 3, the two ways based on the hidden generator point will be investigated, with which we can see either way will carry on the protection works. In section 4,

there is a new algorithm introduced, which is a method with minimizing Hamming weight based on curve cryptography over GF (2^m). In the final section, a conclusion of this paper will be given.

II. Brief Picture for a Traditional ECC Protocol

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \qquad (1)$$

where a, b, c, d, and e, are real numbers.

A special addition operation is defined over elliptic curves and this with the inclusion of a point $\mathcal{O}$, called point at infinity. If three points are on a line intersecting an elliptic curve, then their sum is equal to this point at infinity $\mathcal{O}$, which acts as the identity element for this addition operation. Sometimes the genera equation (1) can be referred as Weierstrass equation as shown in (2):

$$y^2 = x^3 + ax + b \qquad (2)$$

If we wanted use a elliptic curve to be used for cryptography the necessary condition is the curve is not singular, i.e. the discriminant of polynomial $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \qquad (3)$$

Figures 1 and 2 show the two elliptic curves are

$$y^2 = x^3 + 2x + 5 \qquad (4)$$

and

$$y^2 = x^3 - 2x + 1 \qquad (5)$$

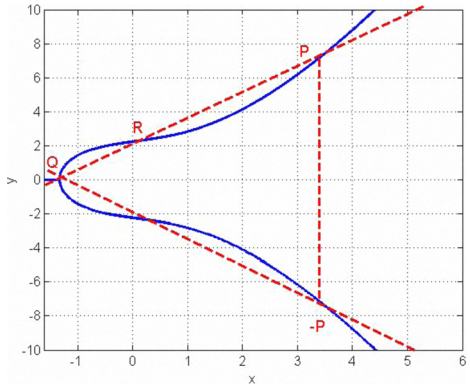

Figure 2. Elliptic curves equation (4)

An elliptic group over the Galois Field $E_p(a,b)$ is obtained by computing $x^3 + ax + b \bmod p$ for $0 \le x < p$. The constant a and b are non negative integers smaller than the prime number p and as here we used "mod p", so equation (3) should be read as:

$$4a^3 + 27b^2 \bmod p \neq 0 \qquad (6)$$

For each value of x one needs to determine whether or not it is a quadratic residue. If it is the case, then there are two values in the elliptic group. If not, then the point is not in the elliptic $E_p(a,b)$ group.

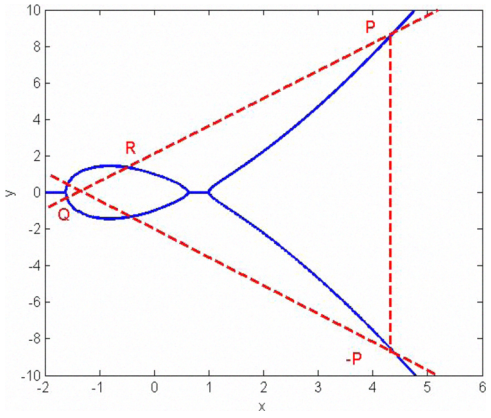

Figure 3. Elliptic curve eqaution (5)

When we fixed a prime number, p and then via the fixed constants a and b we have the Galois Field $E_p(a,b)$ group.

For example, let the points $P = (x_1, y_1)$ and Q (x_2, y_2) be in the elliptic group $E_p(a,b)$ group and $\mathcal{O}$ be the point at infinity. The rules for addition over the elliptic group $E_p(a,b)$ are :

(i) $P + \mathcal{O} = \mathcal{O} + P = P$

(ii) If $x_2 = x_1$ and $y_2 = -y_1$, that is $P(x_1, y_1)$ and $Q = (x_2, y_2) = (x_1 - y_1) = -P$, that is the case: $P + Q = \mathcal{O}$.

(iii) If $Q \neq -P$, then their sum $P + Q = (x_3, y_3)$ is given by ;

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p \qquad (7)$$

where $$\lambda \equiv \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P = Q \end{cases} \qquad (8)$$

In order to describe our new protocol of the hidden generator point we, without losing generality, use an example for the above description. Let's have $p = 23$ (in real case the p will be much larger than this) and $a = 1$ and $b = 1$, i.e. the equation becomes: $y^2 = x^3 + x + 1 \bmod 23$. We have $4a^3 + 27b^2 \bmod 23 = 8 \neq 0$. Now we need to determine if y_2 is in the set of quadratic residues or not. The calculation results are shown below for the elliptic group $E_p(a,b) = E_{23}(1,1)$ which includes the point (4, 0) corresponding to the single value $y = 0$.

The elliptic curve cryptography can be used to encrypt plaintext messages, M, into ciphertexts. The plaintext message M is encoded into a point P_M from the finite set of points in the elliptic group, $E_p(a,b)$. First step consists in choosing a generator point, $G \in E_p(a,b)$, such that the smallest value of n for which $nG = \mathcal{O}$ is a very large prime number. Normally the traditional ECC protocol is let the elliptic group $E_p(a,b)$ and the

generator point G be in public. The each user select a private key, say $n_A < n$ and compute the public key P_A as $P_A = n_A G$. Then, encrypt the message point P_M for the partner, say from Alice to Bob. So Alice (A) chose a random integer k and computes the ciphertext pair of points P_C using Bob's public key P_B:

$$P_C = [(kG), (P_M + kP_B)] \quad (9)$$

Bob received the ciphertext pair of points, P_C then multiplies the first point, (kG) with his private key, n_B, and then adds the result to the second point in the ciphertext pair of points as shown below:

$$(P_M + kP_B) - [n_B(kG)] = P_M \quad (10)$$

which is the plaintext point, corresponding to the plaintext message M. It is noted that only Bob can obtain retrieve the plantext information P_M by the private key n_B. The cryptographic strength of ECC lies in the difficulty for a cryptanalyst to determine the secret random number k from kP and P itself. The fast method to solve this problem is known as the elliptic curve logarithm problem (ECLP) [16].

III. Our Proposal Methods Protecting from Man-in-the Middle in ECC

We have seen that the ECC did not take care of the man-in-the middle attacks even ECC itself has its cryptographic strength as described above.

As above shown that the generator point G and elliptic group $E_p(a,b)$ are in public. Now let's have a closer look at the elliptic group $E_p(a,b)$. In our above example, we pick the prime number $p = 23$ (it is noted that this is only for explaining the new protocol, in real life the p is bigger than this), we have quadratic residues group $(p - 1)/2 = 11$ and for this group the $E_p(a,b)$ can be shown as below:

$$E_{23}(1,1) = \begin{Bmatrix} (0,1) & (0,22) & (1,7) & (1,16) & (3,10) & (3,13) & (4,0) \\ (5,4) & (5,19) & (6,4) & (6,19) & (7,11) & (7,12) & (9,7) \\ (9,16) & (11,3) & (11,20) & (12,4) & (12,19) & (13,7) & (13,16) \\ (17,3) & (17,20) & (18,3) & (18,20) & (19,5) & (19,18) & \end{Bmatrix} \quad (11)$$

As we described in above that any point sitting in equation (11) can be appointed as generator point "G," in the traditional way (as in section II) the G is fixed and let it be in public. But now we are not going to do so. As the generator is hidden, there is no way to know which point is generator therefore the attacker can not make the "man-in-middle" attack. Now we are going to show two ways to complete the ECC processing, first way is that making protocol that has the common principle to work out the generator point, say from the distribution of elliptic $E_p(a,b)$ group; or by the new protocol to work out the P_M as shown below.

In order to make a common principle to work out the generator point for Alice and Bob, say we are going to use the distribution of elliptic $E_p(a,b)$ group, we need to check the what it looks like. The equation (11) can be shown in Figure 4.

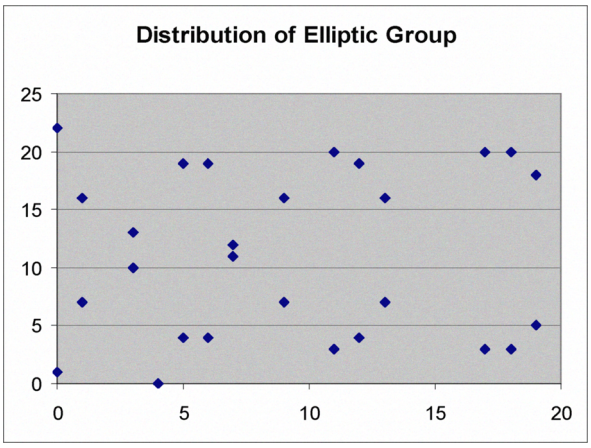

Figure 4. Distribution of Elliptic Group $E_{23}(1,1)$.

We now pick a generator point G by a character of the above distribution, say we pick the G when $G\ (a, b) \in E_p(a,b)$ with a = max $\{a\}$ and b = max $\{b\}$ (it is noted that other principle will apply). In this example, G = (19, 18). Note that we put a = max $\{a\}$ first, so choosing it a = 19 then choose b = max $\{b\}$. The order is important, in this case it is not the G = (18, 20). When the generator point fixed we can have the following processing as described in section II. In fact this way is more secure as the man (or women) has to do is decrypt the message from Bob re-encrypt it with Alice's key and he can monitor the communication without detection. This situation can be shown in Figure 5.

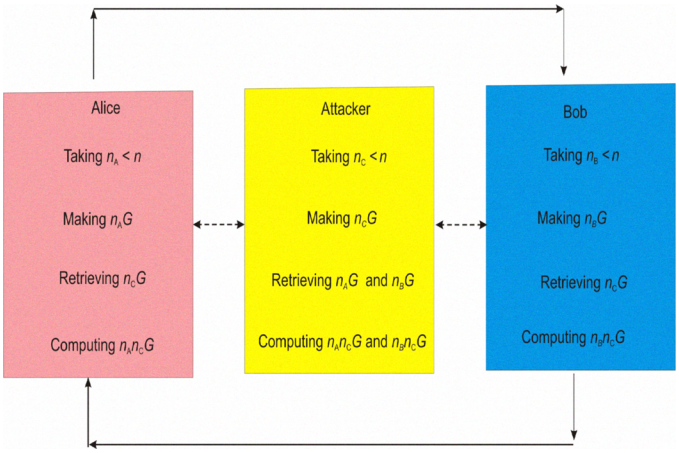

Figure 5. A new protocol protecting the man-in-the-middle attck. As the G is not at the public, the attack cannot work out nCG as tranditional way does, so even the attacker can mornite the communication but no way to understand and attack the communications. The yellow one does not work for the new protocol and the real line works with a protecting.

This issue involves the user of a trusted "certificate authority" (CA). When is queried the CA and returns a digitally signed "certificate" that can be compared to one that has been transmitted by another means. In an authenticated key exchange based on the difficulty of the k^{th} root problem was described in section II.

Now let's turn to our 2nd way, i.e., a new protocol to get the hidden generator point done.

In the 2nd way, we need to face the case that Alice has no information about Bob's public key as traditional way does. Therefore if Alice would like to send a message to Bob, Alice cannot use public key to make cryptography to the message Alice wanted to send. We may use, as an example, a protocol shown in Figure 6.

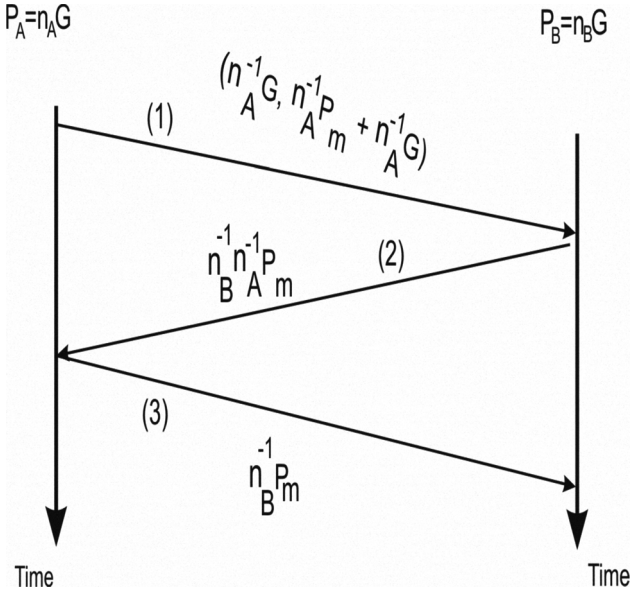

Figure 6. A protocol for the ECC with hidden generator point.

When Alice is going to send the message to Bob, Alice sends the pair of points P_C (as shown (1) in the figure) as below:

$$P_C = [(n_A^{-1}G), (n_A^{-1}P_M + n_A^{-1}G)]$$

Here, $n^{-1}{}_A$ meets the equation: unity = $n^{-1}{}_A\, n_A$, we still called $n^{-1}{}_A$ as private key for Alice but there is no need to worry about the public key as G is hidden at current situation. So either P_A or P_B is not really useful in this case. When Bob received P_C, he can operate as below:

$$n_A^{-1}P_M = n_A^{-1}P_M + n_A^{-1}G - n_A^{-1}G$$

Then, Bob can make P_D as below and sends it to Alice as shown the (2) in Figure 5.

$$P_D = n_A^{-1}n_B^{-1}P_M$$

When Alice received P_D, Alice can make P_E and sent it to Bob as shown (3) in the Figure 5.

$$P_E = (n_A)P_D = (n_A)(n_A^{-1}n_B^{-1}P_M) = n_B^{-1}P_M$$

Then when Bob received P_E, Bob can obtain the message related P_M that sent from Alice by

$$P_M = n_B n_B^{-1}P_M$$

This can obtained only by Bob as no one has the private key that Bob has.

It is clear that the first way discussed above is less computing calculation in comparison with the second way but it is need the "common principle" or "common protocol" before the communication. If this common protocol is to be sent by communication network it will have a risk to be attacked or it will have to create a "safe way" to inform first then go ahead for the rest. For the second way, it is obviously it takes more time than that in traditional way, which is the price to pay for protecting communications from the man-in-middle attacks.

IV. SPEEDING UP ALGORITHMS OVER ECC

Experiences showed that for the popular and powerful ECC, the most expensive operation in elliptic curve based cryptographic protocol is the scalar multiplication. There are many papers investigated this issue, such as ECC using modified complementary [17], binary method [18], non-adjacent form (NAF) [19], and mutual opposite form (MOF) [18] and complements method [20], etc.

The scalar multiplication is very expensive operation in elliptic curve based cryptographic protocol. Hence, the speed of scalar multiplication plays an important role in the efficient system.

Scalar multiplication is the computation of the form $Q = kP$, where P and Q are the elliptic curve points (as figures shown) and k is an integer. It can be obtained by repeated elliptic curve point addition and doubling operations. In the binary algorithms, the integer k is represented as

$$k = \sum_{j=0}^{l-1} k_j 2^j \text{ where } k_j \in \{0,1\}, \qquad (12)$$

which scans the bits of k either from left-to-right or right-to-left. The cost of multiplication depends on the length of the binary representation of k and the number of Harming weight of scalar representation in this representation. If the representation $(k_{n-1}\ldots k_0)_2$ with $k_{n-1}\neq 0$ then the number of doubling operation is $(n-1)$. In an average, binary algorithm requires $(n-1)$ doublings and $(n-1)/2$ additions. For example, k= 1778, then k= $(11011111100)_2$ so computation of 1778 P requires 10 doublings and 5 additions.

It is well known that a algorithm called non-adjacent form (NAF), based on the fact that k is represented as $k = \sum_{j=0}^{l-1} k_j 2^j$, with $k_j \in \{-1,0,1\}$, which using three digits {0, 1, -1}-radix 2 representation and this conversion is taken from right-to-left. The average Hamming weight of signed binary representation is $n/3$ and it has the lower Hamming weight than the binary algorithm. However, it is noted the Hamming weight is one of keys to handle computation load, for example, k = 255, or $(11111111)_2$, computation of 255P requires 7 point additions, but if it is transformed by (10000000-1)P, which is 256P-P, only one addition is required.

There is another algorithm needs to be mentioned, the mutual opposite form (MOF), which converts the binary string to MOF from the most significant bit efficiently. The n-bit binary string k is converted into a signed binary string by $mk = 2k - k$, with "–" stands for a bit subtraction. The conversion of MOF representation of an integer is highly flexible because conversion can be made either from right-to-left or left-to-right. The output of MOF is comparable efficiency with out of NAF as shown in.

As above described it is clearly to see that every mentioned algorithm makes the target that decreasing the Hamming weight to increase the efficient computation over ECC. As shown that the MOF and complementary algorithms are almost the same level in terms of computation costs we may take complementary algorithm as part of hybrid algorithm as shown below. But we need to present the so-called "the 1's complement of binary numbers" described by Gillie [21].

The 1's complement of any binary number may be found by the following equation [21]:

$$C_1 = (2^a - 1) - k \tag{13}$$

Or $$k = (2^a - 1) - C_1 \tag{14}$$

where C_1 = 1's complement of the number

a = number of digits to be handled by the computer

k = binary number whose 1's complement

As an example, let k =1788, or k = $(11011111100)_2$ in its binary form. C_1 = 1's Complement of the number of k and the a in this example it is in binary form is 11. Therefore from the equation (12) we have:

$$C_1 = (2^a - 1) - k = (2^{11} - 1) - (11011111100)$$
$$= 00100000011$$

Therefore we

$$k = 1788 = (2^a - 1) - C_1$$
$$= (2^{11} - 1) - 00100000011$$

This means k = 1788 = $(100000000000 - 00100000011 - 1)_2$,

which gives :

$$1788 = 2048 - 256 - 2 - 1 - 1$$

Hence, this shows that the Hamming Weight of scalar k has reduced from original 8 to current 5 which will save 3 elliptic curve addition operations. One addition operation requires 2 Squaring, 2 Multiplication and 1 inverse operation. But if the original binary form of k is critical for this method as if the number of 1s in original binary form of k is > the one-half of the bit's length, i.e. 1's number $\geq a/2$ then there is no need to convert the original binary format into "1's complement format" as our target is to decrease Harming weight. So our proposed algorithm is that (1) check the 1's number of the binary form, if it is $\geq a/2$, then go to the "complementary algorithm", if it is not, then go to "1's complement format", i.e. go to the equation (13) then go to equation (14).

It is noted that here there is a checking processing before go head for which way to calculate the scale multiplication, which there is time costs but as the either way for the computation of the scale multiplication is the most efficient due to the minimizing Harming weight the saved time can pay the checking costs. In fact the final results, which is shown in the next section by the table, support this conclusion due to the checking processing is almost costing nothing in comparison with the saved time when the Harming weight is minimized.

V. CONCLUSION

We have introduced the secure effective and efficient sensor networks in this paper.

As security issue has been paid more attentions. In recent years some cryptographic algorithms have obtained popularity due to properties that make them suitable for use in constrained environment such as mobile information appliances, sensor networks, where computing resources and power availability are limited. Elliptic curve cryptography (ECC) is one of them. However, in the applications of ECC, in particular for the sensor networks there are always so-called man-in–middle attacks, in particular those networks are with very limited computing capacity and restricted power resources, which drew the researchers' attractions. In this paper we have presented two methods for protecting from man-in-middle attacks based on hidden generator point with ECC.

The importance of security in communication system has become increasingly prominent, and its major technology cryptography technology develops rapidly. ECC has become an important branch of public key cryptography system as it has many benefits for the devices for wireless network, which has restrictions of the limited bandwidth, processing power, and storage space and power consumption.

The efficiency of ECC implementation is highly dependent on the performance of arithmetic operations of scalar multiplication. This paper based on discussions of the current major algorithms present a novel algorithm, hybrid of the "1's complement of binary number" and "complementary" to minimizing Harming weight to speed up the calculation over ECC. The final results are summarized in the table shown in below, where the results obtained from [15] was used.

In terms of average, the proposed algorithm is about 12.5% saved time in comparison with the results of complementary algorithm in [17].

As we have seen that due to the checking processing, there is always the case that the Hamming Weight will less than the half of the length (in terms of digit number) and the either complement of the number method or 1's complement of the number method will constantly keep the Hamming Weight minimizing, which makes this method sitting on the very power saving position due to the computing works.

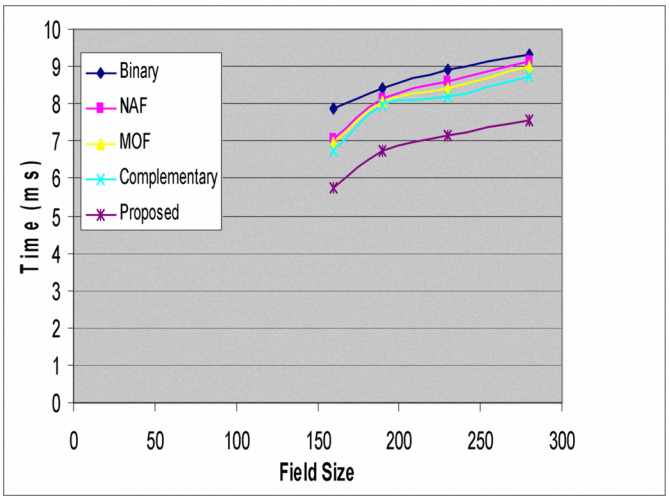

Figure 7. Comparison the proposed algorithm with the nominated algorithms by [17] (the data for the nominated algorithms were used from the same reference).

REFERENCES

[1] A. Silberstein, G. Puggioni, A. Gelfand, K. Munagala, and J. Yang, "Supperession and failures in sensor networks: a bayesian approach," In: VLDB 2007, pp842-853. VLDB Endowment (2007).

[2] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," Communications of the ACM 13, 422-426 (1970).

[3] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks," In: SIGMOD 2006, pp157-168. ACM, New York (2006).

[4] Heejung Yang and Chin-Wan Chung, "An effective and efficient method for handling transmission failures in sensor networks," DASFAA 2009, LNCS 5463, pp92-106, 2009. Springer-Verlag Berlin Heidelberg 2009.

[5] D. Bernstein, "High-speed diffie-hellman, part 2," presented at the INDOCRYPT'06 tutorial session, Dec. 11-13, Kolkata, India (2006).

[6] J. Adikari, V. Dimitrov, and L. Imbert, "Hybrid binary-ternary joint sparse from and its application in ellipic curve cryptography," Cryptology ePrint Archive, Report 2008/285, 2008.

[7] Bangju Wang, Huanguo Zhang and Yuhua Wang, "An efficient elliptic curves scalar multiplication for wireless network," 2007 IFIP International Conference on Network and Parallel Computing-Workship, pp131.

[8] Shiwei Ma, Yuanling Hao, Zhongqiao Pan, and Hui Chen, "Fast implementation for modular inversion and scalar multiplication in the elliptic curve cryptography," 2008 Second International Symposium on Intelligent Information Technology Application, pp488.

[9] Michael Scott, "Optimal Irreducible Polynomials for GF(2^m) Arithmetic," Cryptology ePrint Archive, Report 2007/192, 2007.

[10] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," International Journal of Security and Networks, vol. 1, no.3/4, 2006.

[11] A. Liu, P. Kampanakis, and P. Ning, "TinyECC: Elliptic curve cryptography for sensor networks ," (version 10), november 2007.

[12] Alicia Nicki Washington and Rotimi Iziduh, "Modeling of military networks using group mobility models,"2009 sixth International Conference on Information Technology: New Generations," pp1670

[13] Kyoung-Lae Noh, Yik-Chung Wu, Khalid Qaraqe, and W. Suter, "Extension of pairwise broadcast clock synchronization for multicluster sensor networks," Hindawi publishing Corporation, EURASIP Journal on Advances in Sigtnal Processing Vol. 2008, pp1-10.

[14] Liqian Luo and Michael Ward, "Design, implementation, and evaluation of enviroMic: a storage-centric audio sensor network," ACM Transactions on Sensor Networks, Vol. 5 No. 3, Article 22, May 2009, pp22:1-22:35.

[15] Elyes Ben Hamida and Guillaume Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," IEEE Wireless Communications 15, 6 (2008) 31-37.

[16] W. Stallings, "Cryptography and Network Security: Principles and Practice," Prentice-Hall, Upper Saddle River, New-Jersey, second edition, 1999.

[17] P. Balasubramaniam and E. Karthikeyan, "Elliptic curve scaslar multiplication algorithm using complementary recoding," Applied Mathematics and Computer, 2007 pp.1-6. doi: 10.1016/j.amc.2007.01.015.

[18] Standard Specifications for Public Key Cryptography, IEEE standard 1363, 2000.

[19] Shiwei Ma, Yuanling Hao, Zhongqiao Pan, and Hui Chen, "Fast implementation for modular inversion and scalar multiplication in the elliptic curve cryptography," 2008 Second International Symposium on Intelligent Information Technology Application, pp488.

[20] K. Okeya, "Signed binary representations revisited," Proceedings of CRYPTO'04 (2004) pp123-139.

[21] Angelo C Gillie, "Binary Arithmetic and Boolean algebra," McGRAW-HILL Book Company, 1965. pp53.